Fig. 1A (Prior Art)

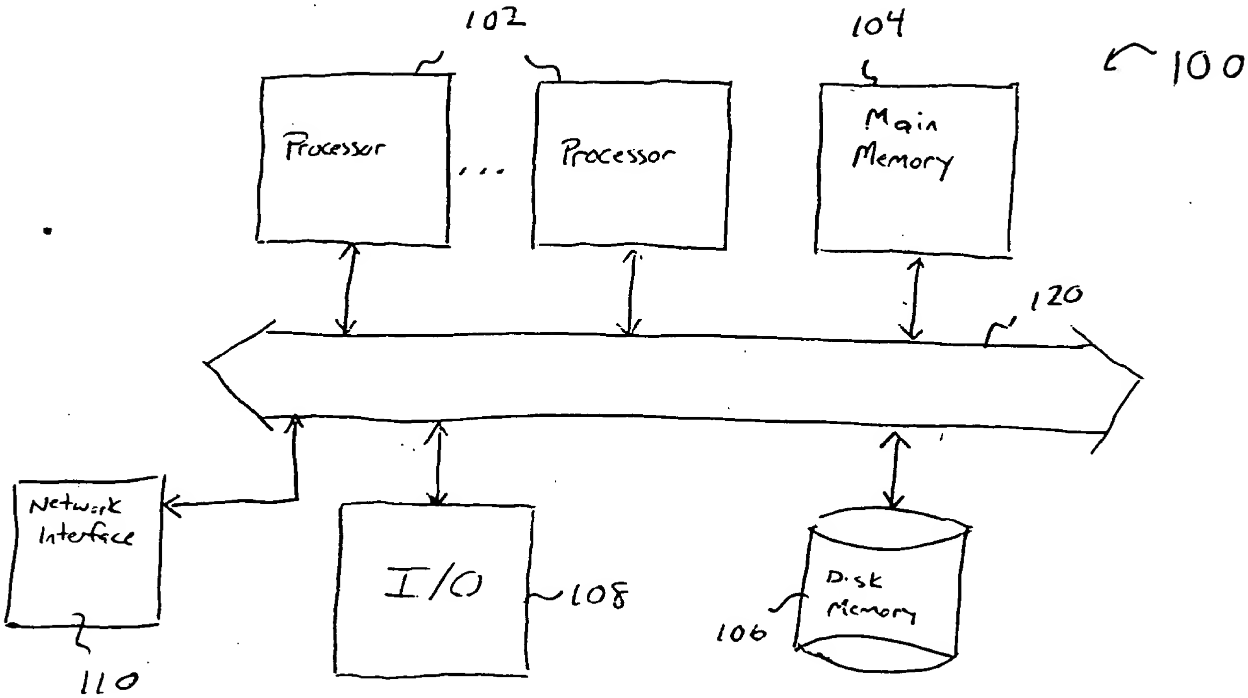


Fig. 2 (Prior Art)

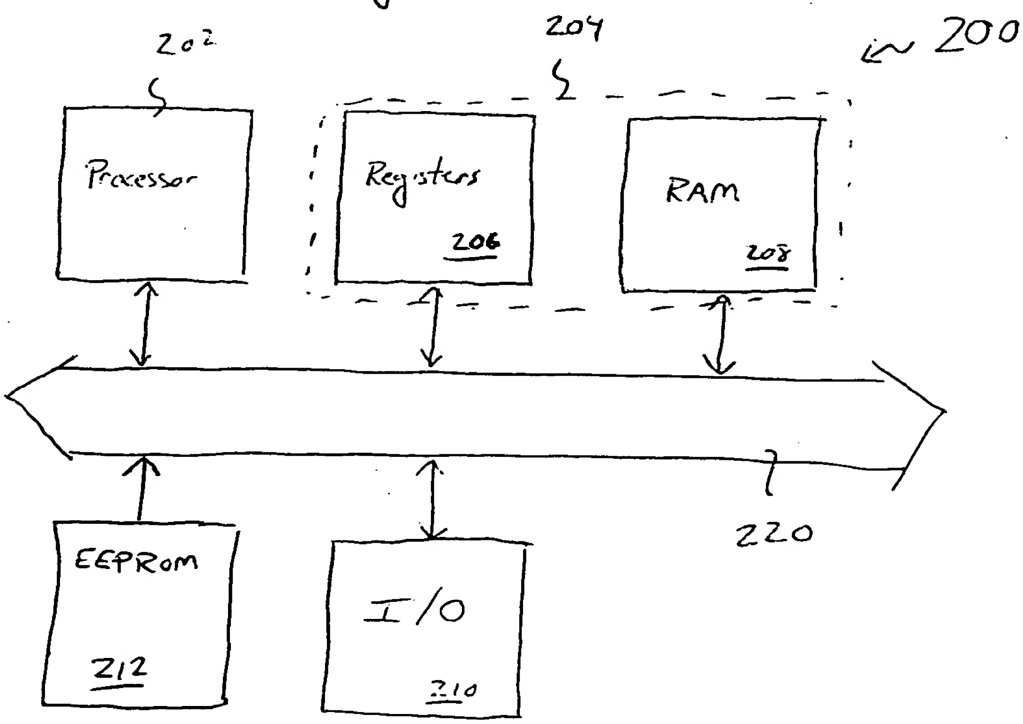


Fig. 1B (Prior Art)

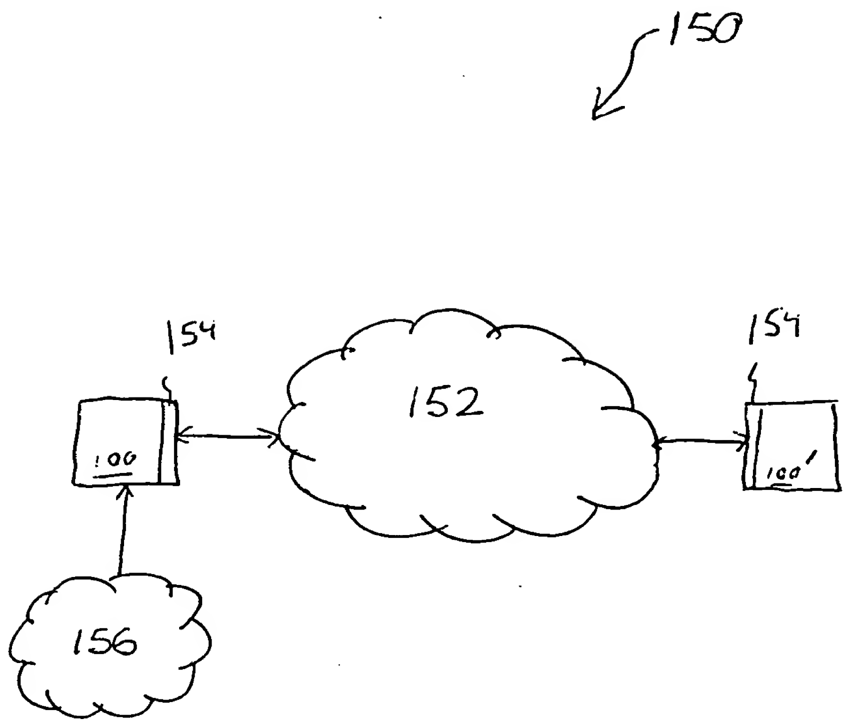


Fig. 3

Step	Party i	Party j
(1)	Party i's cryptography device generates a correct signature E for message m and transmits E to party j's cryptography device.	
		Party j's cryptography device receives E and stores it in memory.
(2)	Party i's cryptography device generates incorrect signature $\hat{E}$ for the same message m and transmits $\hat{E}$ to party j's cryptography device.	
		Party j's cryptography device receives $\hat{E}$ and stores it in memory.
(3)		Party j's cryptography device determines $a(E_1-\hat{E}_1)$; $\gcd(E-\hat{E},\ N)=q$.
(4)		Having determined q, party j's cryptography device determines N.

Fig. 4

Step	Party i	Party j
(1)	Party i's cryptography device generates erroneous signature $\hat{E}$ for known message m (i.e., message m is generated without padding or with non-random padding) and transmits $\hat{E}$ to party j's cryptography device.	Party j's cryptography device receives $\hat{E}$.
(2)		Party j's cryptography device determines $\gcd(M-\hat{E}^{ei}, N)=q$.
(3)		Having determined q, party j's cryptography device determines N.

Fig. 5

Step	Party i	Party j
(1)	Party i's cryptography device selects a random r, generates $r^2 \bmod N$, and transmits this value to party j's cryptography device.	
		Party j's cryptography device observes the value $r^2 \bmod N$.
(2)		Party j's cryptography device generates a random subset $S \subseteq (1, \ldots, t)$ and transmits S to party i's cryptography device.
(3)	While waiting to receive S from party j's device, a value in party i's device is inverted. After receiving S from party j's device, party i's device generates $\hat{y} = (r+\acute{E})\Pi_{i \in S} s_i$. Party i's device transmits $\hat{y}$ to party j's device.	
		Party j's device receives $\hat{y}$.
(4)		Party j's device determines $\acute{E}$ by finding an $\acute{E}$ satisfying $(r+\acute{E})^2 = \frac{\hat{y}^2}{\Pi_{i \in S} v_i} \pmod N$
(5)		This is possible because $\acute{E} = 2^k$ for some $1 \le k \le n$
		Party j's device may determine r using:

Fig. 5 Con't

$$(r+\acute{E})^2 - r^2 = 2\acute{E}r + \acute{E}^2 \pmod N$$

6 (~~7~~)

Party j may use r to determine $\Pi_{i\epsilon S} s_i$ using:

$$\prod_{i\epsilon S} s_i = \frac{2\acute{E}\hat{y}}{\frac{\hat{y}^2}{\prod_{i\epsilon S} v_i} - r^2 + \acute{E}^2} \pmod N$$

7 (~~8~~)

Party j's cryptography device determines $\Pi_{i\epsilon S} s_i$ for various sets U_i construced as either (1) singleton sets or (2) selected at random such that resulting characteristic vectors are linearly independent.

8 (~~9~~)

Using the sets constructed above, party j's device performs the Fiat-Shamir authentication scheme, providing the sets to party i's device.

(9)

Using the response to the sets sent to party i's device, party j determines the secret values $s_1, \ldots, s_t$.

?

Fig. 6A (Prior Art)

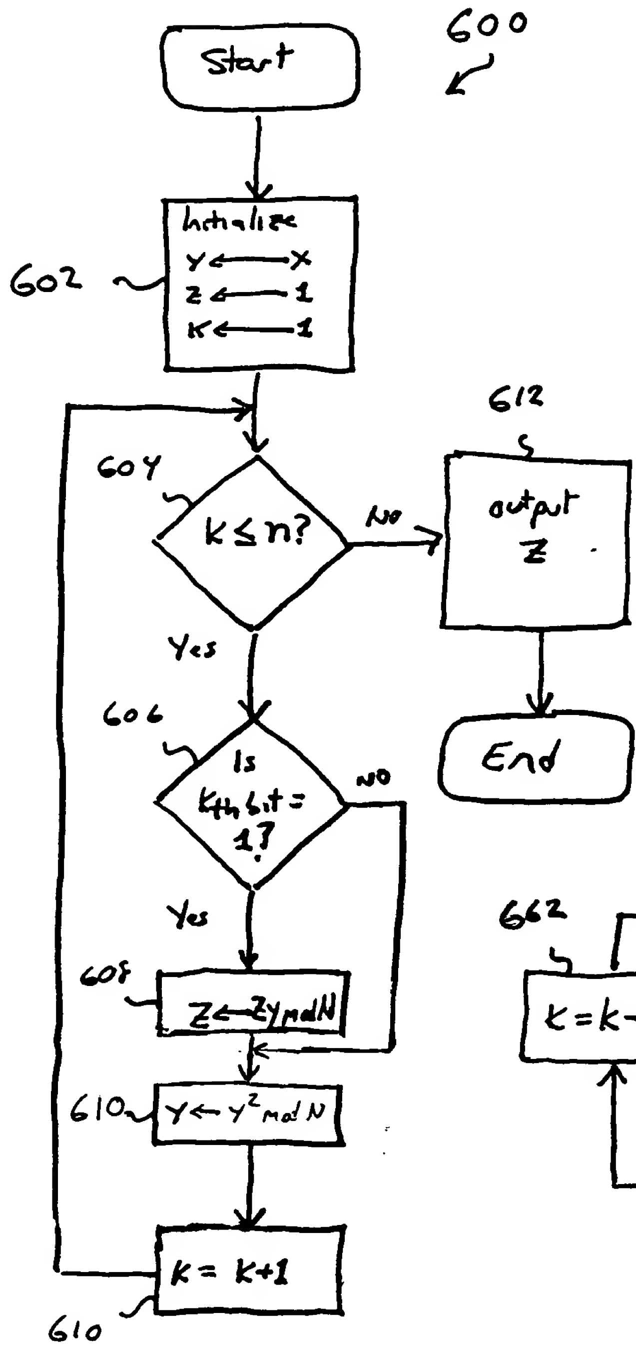


Fig. 6B (Prior Art)

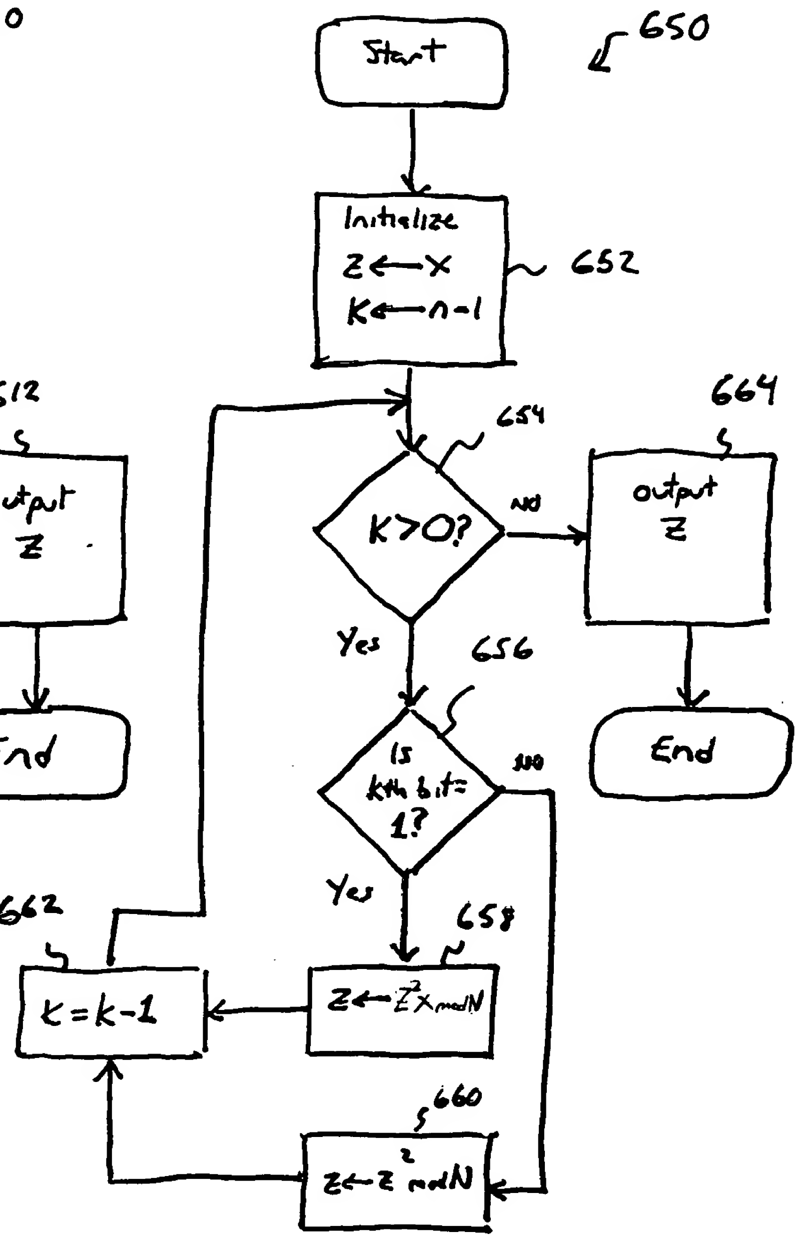
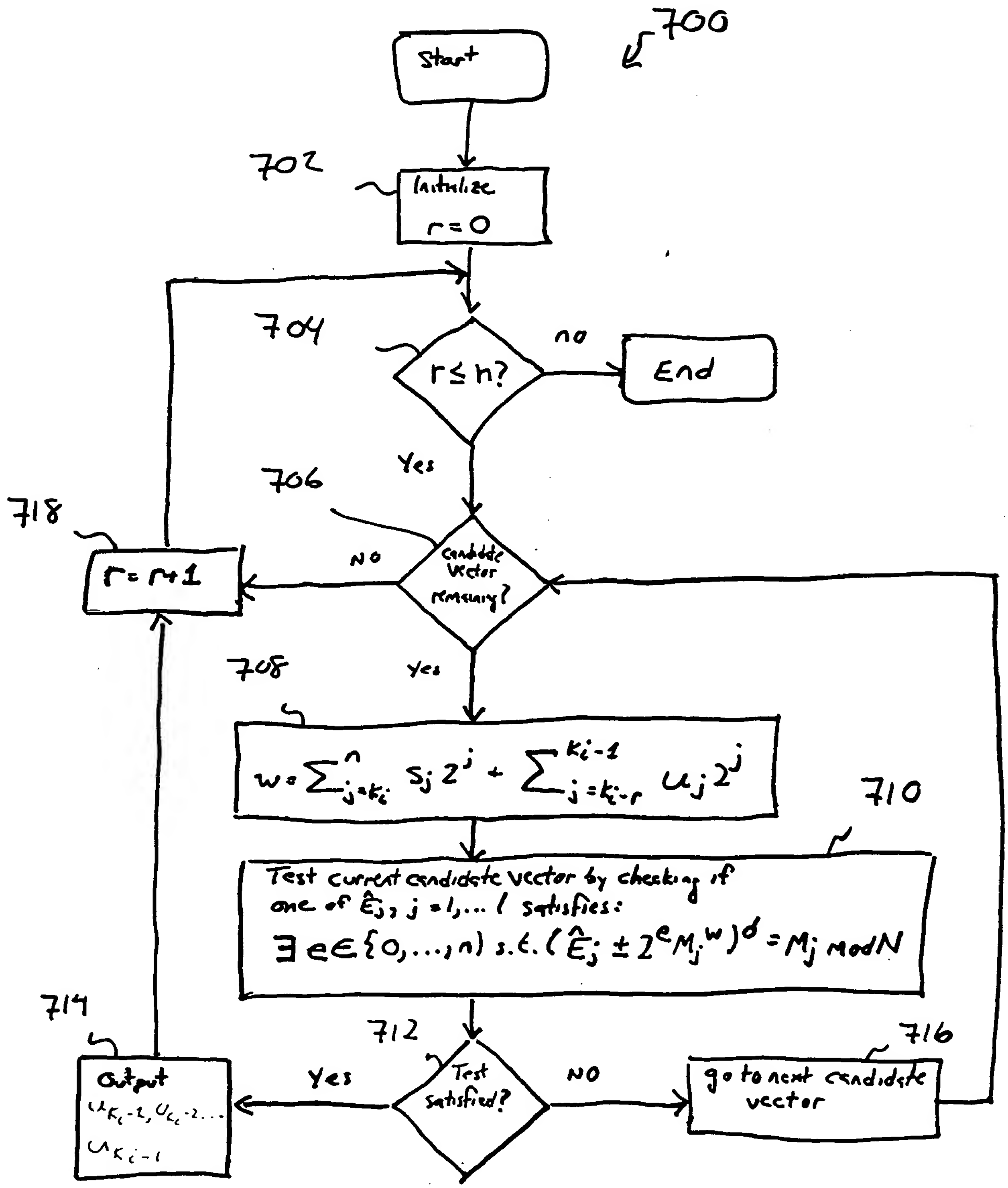


Fig. 7


700
Start
702
Initialize r=0
704
r≤n?
no
End
Yes
706
Candidate vector remaining?
NO
718
r=r+1
yes
708
$w = \sum_{j=k_i}^{n} s_j 2^j + \sum_{j=k_i-r}^{k_i-1} u_j 2^j$
710
Test current candidate vector by checking if one of $\hat{E}_j$, j = 1,... l satisfies:
$\exists e \in \{0,\dots,n\}$ s.t. $(\hat{E}_j \pm 2^e M_j^{w})^d = M_j \bmod N$
714
Output $u_{k_i-1}, u_{k_i-2} \dots u_{k_i-r}$
Yes
712
Test satisfied?
NO
716
go to next candidate vector